

OpenCV & Python

Основы работы с изображениями

ВВОД / ВЫВОД

`imread(filename[, flags]) -> retval`

- ❑ `IMREAD_COLOR`
- ❑ `IMREAD_GRAYSCALE`
- ❑ `IMREAD_UNCHANGED`

`imshow(winname, mat) -> None`

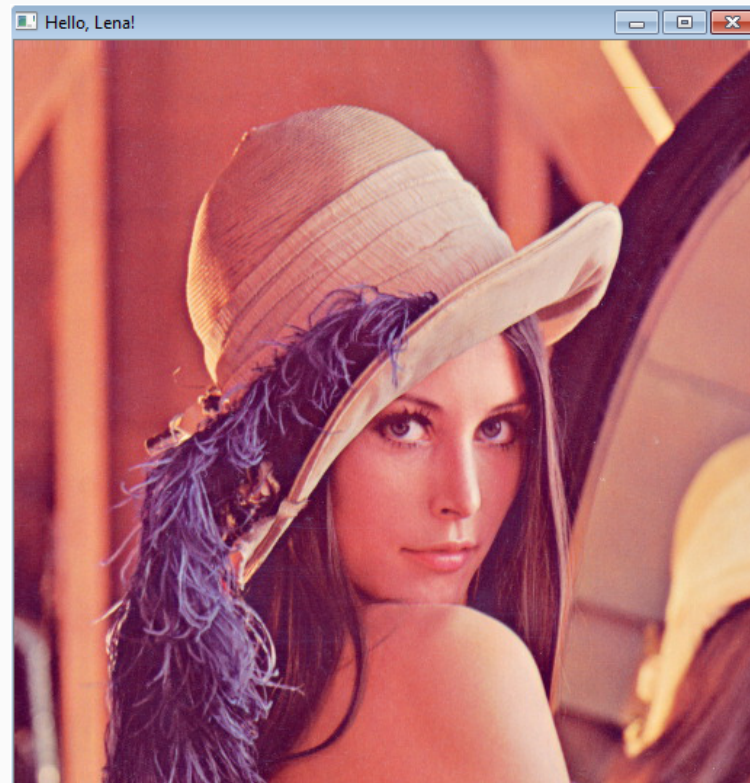
`imwrite(filename, img[, params]) -> retval`

Пример: I/O. "Hello, Lena!"

```
import cv2

lena = cv2.imread("Lena.jpg")
cv2.imwrite("LenaCopy.jpg", lena)
cv2.imshow("Hello, Lena!", lena)

cv2.waitKey(0)
```



Свойства изображения

`image.shape` - кортеж, включающий число строк, столбцов и каналов (если изображение цветное)

`image.size` - количество байт, выделенных под изображение

`image.dtype` - используемый тип данных (обычно, `dtype=np.uint8`)

Пример: Свойства изображения

```
import cv2

lena_color      = cv2.imread("Lena.png", cv2.IMREAD_COLOR)
lena_unchanged  = cv2.imread("Lena.png", cv2.IMREAD_UNCHANGED)
lena_grayscale  = cv2.imread("Lena.png", cv2.IMREAD_GRAYSCALE)

print(type(lena_color))           # <class 'numpy.ndarray'>
print('RGB shape: ', lena_color.shape) # RGB shape: (512, 512, 3)
print('ARGB shape:', lena_unchanged.shape) # ARGB shape: (512, 512, 4)
print('Gray shape:', lena_grayscale.shape) # Gray shape: (512, 512)
print('lena.dtype:', lena_color.dtype) # lena.dtype: uint8
print('lena.size: ', lena_color.size) # lena.size: 786432
```

Работа с пикселями и прямоугольными областями изображения

`image[x, y]` - пиксель с координатами (x, y)

`image.item(*args)` - копия элемента массива

`image.itemset(*args)` - присваивание элементу массива

`image[x]` - x -ая строка `image`

`image[:, y]` - y -ый столбец `image`

`image[0:x, 0:y]` - прямоугольная область площадью $x * y$ пикселей, расположенная в верхнем левом углу `image`

Пример: Работа с областями изображения

```
import cv2

lena = cv2.imread("Lena.png")

n_rows, n_cols, n_channels = lena.shape

lena_top = lena[0:n_rows / 2, :]
lena_left = lena[:, 0:n_cols / 2]

lena_top[:] = lena_top[::-1]
lena_left[:] = lena_left[::-1]

cv2.imshow("Lena Broken", lena)
cv2.waitKey(0)
```



Арифметические операции над изображениями

`add(src1, src2[, dst[, mask[, dtype]]]) → dst`

`subtract(src1, src2[, dst[, mask[, dtype]]]) → dst`

`absdiff(src1, src2[, dst]) → dst`

`multiply(src1, src2[, dst[, scale[, dtype]]]) → dst`

`divide(src1, src2[, dst[, scale[, dtype]]]) → dst`

Пример: Арифметические операции над изображениями

```
import cv2

lena_gray = cv2.imread("Lena.png",
                       cv2.IMREAD_GRAYSCALE)

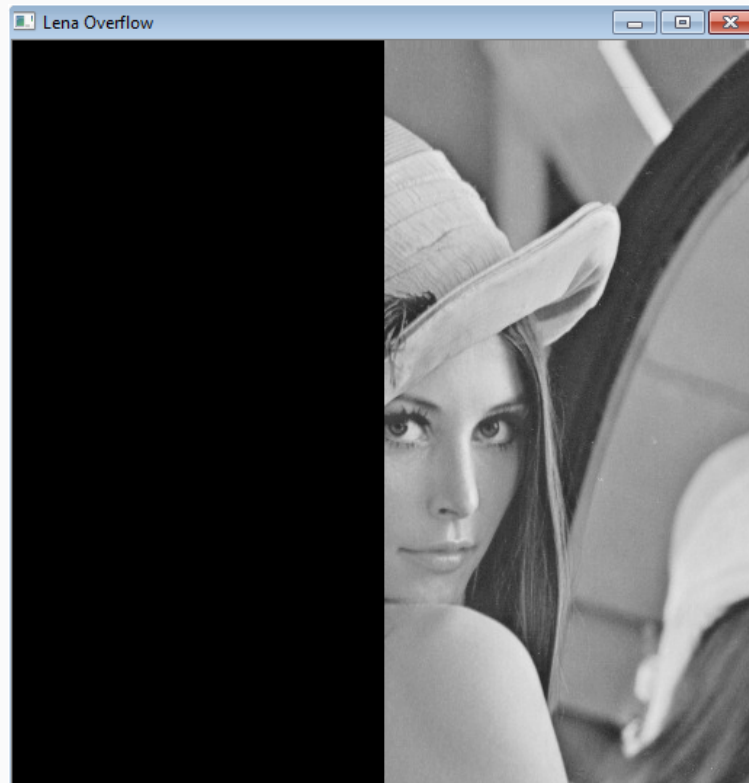
n_rows, n_cols = lena_gray.shape

lena_left_half = lena_gray[:, 0:n_cols / 2]
lena_right_half = lena_gray[:, n_cols / 2:n_cols]

lena_left_half = cv2.subtract(lena_left_half, 256)
lena_right_half = lena_right_half - 256

lena_gray[:, 0:n_cols / 2] = lena_left_half
lena_gray[:, n_cols / 2:n_cols] = lena_right_half

cv2.imshow("Lena Overflow", lena_gray)
cv2.waitKey(0)
```



Побитовые операции над изображениями

`bitwise_and(src1, src2[, dst[, mask]]) → dst`

`bitwise_not(src[, dst[, mask]]) → dst`

`bitwise_or(src1, src2[, dst[, mask]]) → dst`

`bitwise_xor(src1, src2[, dst[, mask]]) → dst`

Пример: Побитовые операции над изображениями

```
import cv2

lena = cv2.imread("Lena.png")

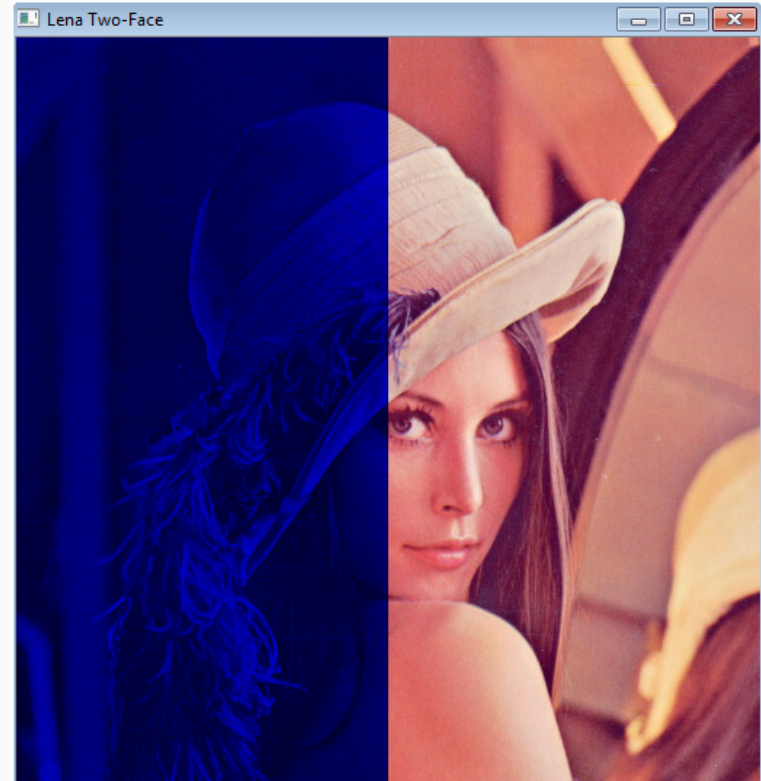
n_rows, n_cols, n_channels = lena.shape

lena_left_half = lena[:, 0:n_cols / 2]
lena_right_half = lena[:, n_cols / 2:n_cols]

lena_left_half = cv2.bitwise_and(lena_left_half,
                                  0xFF)
lena_right_half = lena_right_half & 0xFF

lena[:, 0:n_cols / 2] = lena_left_half
lena[:, n_cols / 2:n_cols] = lena_right_half

cv2.imshow("Lena Two-Face", lena)
cv2.waitKey(0)
```



Доступ к каналам

`image[:, :, channel]`

`split(m[, mv]) → mv`

`merge(mv[, dst]) → dst`

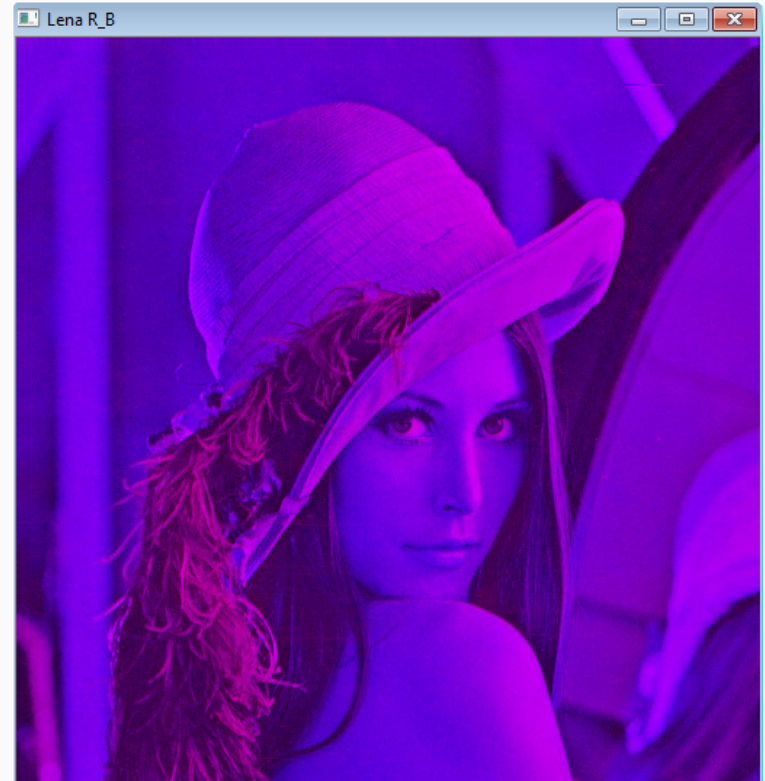
Пример: Доступ к каналам

```
import cv2

bgr = cv2.imread("Lena.png")

# b, g, r = bgr[:, :, 0], bgr[:, :, 1], bgr[:, :, 2]
b, g, r = cv2.split(bgr)
rgb = cv2.merge((r, g, b))
rgb[:, :, 1] = 0

cv2.imshow("Lena R_B", rgb)
cv2.waitKey(0)
```



Изменение цветового пространства

`cvtColor(src, code[, dst[, dstCn]]) → dst`

- ❑ **RGB** ↔ **GRAY**: `COLOR_BGR2GRAY`, `COLOR_RGB2GRAY`, `COLOR_GRAY2BGR`, `COLOR_GRAY2RGB`
- ❑ **RGB** ↔ **HSV**: `COLOR_BGR2HSV`, `COLOR_RGB2HSV`, `COLOR_HSV2BGR`, `COLOR_HSV2RGB`
- ❑ **RGB** ↔ **CIE L*a*b**: `COLOR_BGR2Lab`, `COLOR_RGB2Lab`, `COLOR_Lab2BGR`, `COLOR_Lab2RGB`

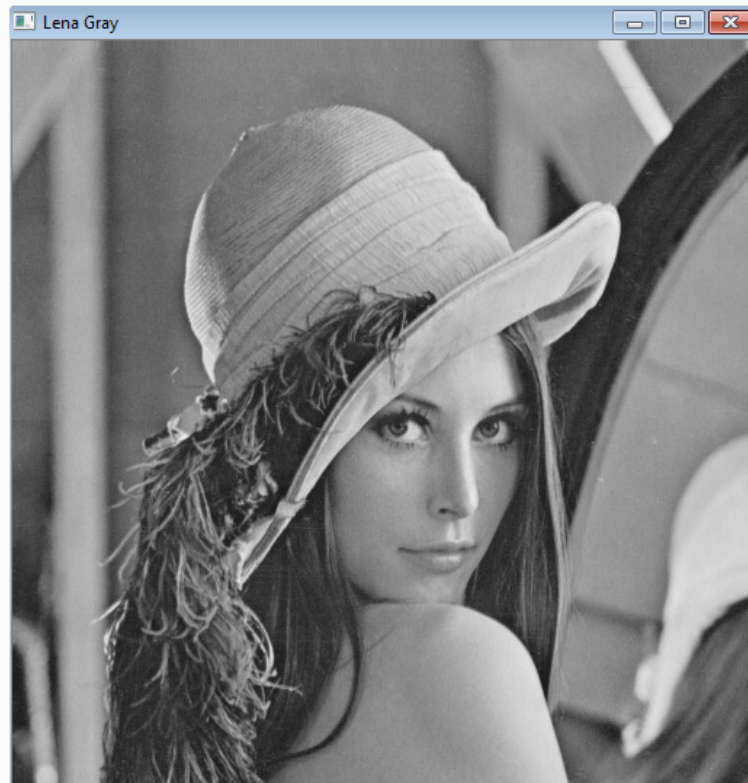
Пример: Изменение цветового пространства

```
import cv2

lena = cv2.imread("Lena.png")

lena = cv2.cvtColor(lena, cv2.COLOR_BGR2GRAY)
lena = cv2.cvtColor(lena, cv2.COLOR_GRAY2BGR)

cv2.imshow("Lena Gray", lena)
cv2.waitKey(0)
```



Полезные ресурсы

- ❏ [Python](#)
- ❏ [Miniconda](#)
- ❏ [StackOverflow - How to install Python OpenCV](#)
- ❏ [OpenCV-Python Tutorials](#)
- ❏ [OpenCV API Reference](#)